ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ
INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT



УДК 004.428+004.94 DOI 10.12737/12593

Особенности реализации механизма подключения библиотек сторонних разработчиков в информационной системе «Канал»*

Н. С. Могилевская[1], К. А. Чугунный[2]**
[1,2]Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Implementation features of the third-party DLL connection mechanism in the information system «Channel»***

N. S. Mogilevskaya[1], K. A. Chugunniy[2] **
[1,2]Don State Technical University, Rostov-on-Don, Russian Federation

Цель работы состоит в создании способа применения механизма динамически подключаемых библиотек при априорно неизвестном количестве и типе параметров экспортируемых функций подключаемых библиотек на примере информационной системы «Канал». Система «Канал» позволяет моделировать цифровые помехоустойчивые каналы связи и решать задачу согласования канала связи и алгебраического метода помехоустойчивой защиты этого канала. Главная особенность системы состоит в том, что использовать ее могут как исследователи, не обладающие навыками программирования, так и специалисты с квалификацией в области программирования вычислительных средств. Первые могут использовать существующий функционал системы, а вторые — дополнительно разрабатывать и подключать к системе собственные библиотеки, расширяющие возможности информационной системы. Описаны проблемы, возникающие при наращивании функционала этой информационной системы сторонними разработчиками методом динамически подключаемых библиотек. Основную сложность составляет априорная неопределенность в названиях, параметрах и типах параметров внешних функций подключаемых библиотек. Метод решения проблемы подключения библиотек сторонних разработчиков состоит в абстрагировании основной программы от данных, используемых подключаемыми модулями, унификации требований к внешним библиотекам, снабжении каждой библиотеки специальным файлом описания и создании для каждой библиотеки процедуры редактирования введенных параметров. Результатом работы является качественная работа системы «Канал» с внешними библиотеками, при соблюдении разработчиками этих библиотек ряда соглашений. Сделаны выводы о том, что полученное решение может быть использовано в других системах.

Ключевые слова: информационная система, ИС «Канал», динамически подключаемые библиотеки, помехоустойчивый канал связи, расширение системы, внешние функции, сторонние разработчики, DLL.

The work objective is to develop a method of the dynamic link libraries mechanism application with aprior unknown number and type of the exported functions parameters of DLL on the example of the information system “Channel”. The system “Channel” allows simulating the noise-immune digital communication channels and solving the problem of matching the communication channel and the algebraic method of the jamproof protection of this channel. The main feature of the system is that it can be used not only by the researchers who do not have programming skills, but also by experts in the computing aids programming. The first-mentioned group can use the existing system software, and experts can further develop and connect their own libraries to expand the capabilities of the information system. Problems that arise when expanding the information system functional by the third-party developers through the DLL technique are described. The main difficulty is the prior uncertainty in the names, parameters, and parameter types of the DLL external functions. The problem-solving technique for the of third-party libraries connection developers is the master program abstraction from the data used by the plugins, the unification of requirements to the external libraries, the supply of each library with a special description file, and the creation of the input parameter editing procedure for each library. The result is a quality operation of the information system “Channel” with the external libraries under observing a number of agreements by the DLL developers. The conclusion is made that the resulting solution can be used in other systems as well.

Keywords: information system, “Channel” IS, dynamic link libraries, noiseless communication channel, system expansion, external function, third-party developers, DLL.

*Работа выполнена в рамках инициативной НИР.
**E-mail: broshka@nm.ru
***The research is done within the frame of the independent R&D.

Введение. Имитационное моделирование помехоустойчивых каналов связи является весьма распространенным и зачастую единственно возможным способом решения сложной многопараметрической задачи теории связи, состоящей в согласовании алгебраического помехоустойчивого кодека и канала связи [1, 2]. Обычно имитационные модели реализуются в виде программных средств. Большинство таких средств создаются для решения одной узкой задачи и используются очень короткое время только самими создателями. Наиболее предпочтительным вариантом реализации имитационной модели являются программные средства с расширяемым функционалом.

Примерами таких систем с расширяемым функционалом являются, например, российская разработка «Имитатор» [3] и система Matlab [4, 5]. Указанные системы обладают рядом несомненных достоинств. Однако с точки зрения их использования для решения задачи согласования кодека и канала эти системы имеют существенные недостатки. Так, расширения программного средства «Имитатор» разрабатываются только ее авторами, а система Matlab требует от всех пользователей навыков программирования.

Информационная система «Канал» [6–8] представляет собой специально разработанное программное средство, позволяющее экспериментально оценивать корректирующие способности помехоустойчивых кодов в условиях различной помеховой обстановки и решать задачу согласования алгебраического помехоустойчивого кодека и канала связи. Ее выгодное отличие от известных аналогов заключается в том, что использовать ее могут как исследователи, не обладающие навыками программирования, так и специалисты с квалификацией в области программирования вычислительных средств. Первые могут использовать существующий функционал системы, а вторые дополнительно могут разрабатывать и подключать к системе собственные библиотеки, расширяющие возможности информационной системы. Наличие механизма подключения библиотек сторонних разработчиков очень важно, так как это дает возможность системе развиваться и быть использованной для решения множества задач.

Подключение новых библиотек к информационной системе можно выполнять с использованием хорошо известного механизма динамически подключаемых библиотек — DLL (Dynamic-Link Library) [9]. Механизм DLL позволяет эффективно наращивать возможности программы за счет программных модулей, реализованных различными разработчиками. Отметим, что исходный код для такой библиотеки может быть создан на многих языках программирования. В тексте DLL-модуля программист располагает множество процедур и функций, реализующих необходимые алгоритмы. Экспортируемые функции, т.е. функции, которые будут использованы внешней программой, указываются с использованием специальной директивы. Основная программа при необходимости загружает в память компьютера DLL-модуль и после этого она может вызывать экспортируемые функции из этого модуля.

Ограничением на использование динамически подключаемых библиотек является тот факт, что программе, вызывающей подключаемую библиотеку, априорно должен быть известен список экспортируемых функций во всех подключаемых модулях, а также число, тип и область допустимых значений параметров вызываемых функций. При реализации информационной системы «Канал» возникает ситуация, при которой сведения о списках параметров априорно не известны. Следовательно, использовать механизм DLL напрямую не удается и требуется разработать модифицированный способ его применения. Рассмотрим ниже проблемы, возникшие при подключении библиотек сторонних разработчиков в ИС «Канал», и найденные методы их решения.

Схема модели канала связи, реализуемая в ИС «Канал». В информационной системе «Канал» [7, 8] механизм подключаемых библиотек используется в ее части, отвечающей за моделирование канала связи. Схема прохождения данных в модели канала представлена рис. 1: данные из источника сообщений обрабатываются помехоустойчивым кодером канала, далее они попадают в канал с помехами, после чего зашумленные данные восстанавливаются в декодере канала и результат передается в приемник сообщений. Некоторый набор алгоритмов, реализующих кодирование и декодирование данных, а также их зашумление распространяется вместе с программным средством, реализующим информационную систему «Канал». Расширение набора этих алгоритмов может быть проведено сторонними разработчиками и выполнено в виде подключаемых модулей. Функционал блоков «Кодер канала», «Линия связи с шумом» и «Декодер канала» может быть реализован в виде подключаемых библиотек.

Отметим, что кодер и декодер канала в рассматриваемой системе могут реализовывать не только один алгоритм кодирования или декодирования, но и представлять собой комбинацию из конечного множества помехоустойчивых кодеков и перемежителей. Такие комбинации называются каскадом. Пример организации каскада представлен на рис. 2. Символами π и π^{-1} на рис. 2 обозначены перемежители и деперемежители, соответственно. Отметим, что пара «Кодер i» и «Декодер i», а также перемежитель «π» и деперемежитель «π^{-1}» должны быть согласованы, т.е. реализовывать связанные алгоритмы.

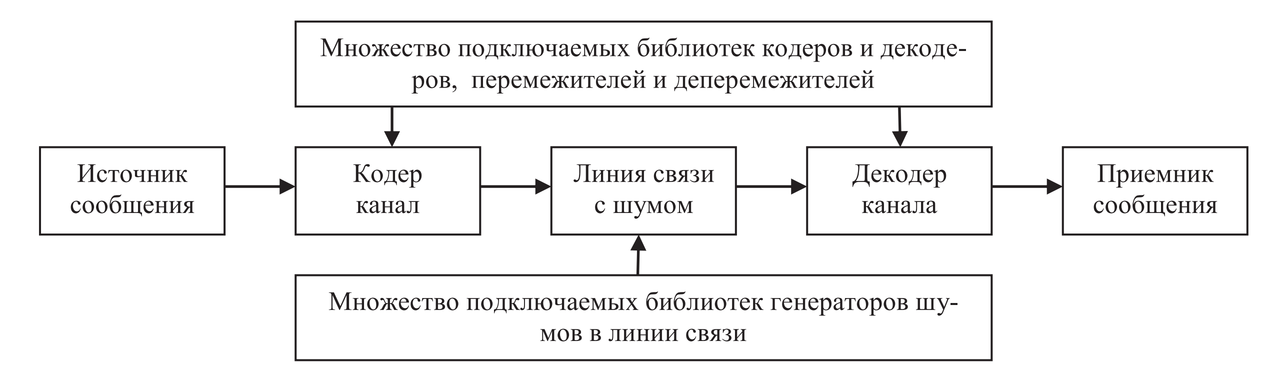


Рис. 1. Схема имитационной модели канала связи

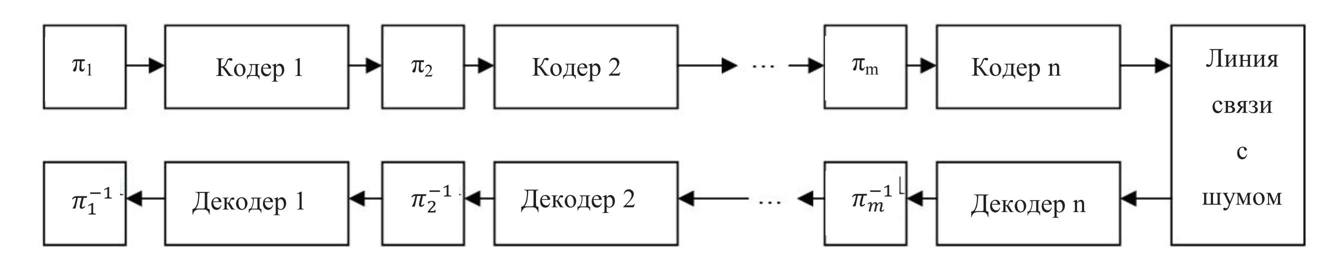


Рис. 2. Пример организации каскада из помехоустойчивых кодеков и перемежителей

Проблема подключения библиотек сторонних разработчиков в ИС «Канал». Как было отмечено выше, в ИС «Канал» рассматриваются три типа подключаемых библиотек в зависимости от их назначения. А именно, библиотека может содержать помехоустойчивый кодек, перемежитель или генератор шума. В зависимости от типа библиотеки ее создателем, согласно технологии DLL, должны быть реализованы определенные интерфейсные функции, обеспечивающие связь с ИС «Канал». Сложность сопряжения этих функций с ИС «Канал» состоит в том, что заранее неизвестны число и типы параметров этих функций, а также область их допустимых значений. Например, в библиотеке, реализующей помехоустойчивый кодек, обязательно должны быть функции кодирования и декодирования. Так, в спецификации функции кодирования двоичным кодом Хемминга [2] достаточно указать только один числовой параметр, который полностью определяет все параметры кода и алгоритма кодирования, а число параметров в спецификации функции кодирования произвольным циклическим кодом [2] зависит от степени порождающего полинома, мощности используемого поля Галуа и может достигать нескольких десятков значений. Аналогичная ситуация широкого диапазона возможного количества и типов параметров функций, вызываемых из подключаемых библиотек, наблюдается и для библиотек, реализующих потоки ошибок и алгоритмы перемежения.

Укажем желаемые свойства искомого технического решения проблемы подключения библиотек сторонних разработчиков.

Самостоятельное подключение новых модулей пользователями и разработчиками. Подключение стороннего модуля, реализующего произвольные алгоритмы кодирования, зашумления или перемежения, вне зависимости от списка параметров функций, экспортируемых этими модулями, должно производиться без взаимодействия с авторами основной программы ИС «Канал».

Простота и удобство создания подключаемых модулей. Очевидно, что чем больше ограничений и требований по отношению к подключаемым библиотекам необходимо учитывать их разработчикам, тем сложнее разработка внешних модулей для программы и тем меньше разработчиков согласиться их создавать. Следовательно, сложность создания подключаемых модулей для ИС «Канал» не должна значительно превосходить сложность программной реализации алгоритмов кодирования, перемежения или зашумления.

Корректное взаимодействие основной программы с подключаемыми модулями. Программа должна корректно работать с подключаемыми кодеками, вызывать их, редактировать параметры, обрабатывать исключительные ситуации.

Удобство для пользователя. С точки зрения пользователя не должна различаться работа с функционалом подключаемых модулей и функционалом основной программы. А именно, должно быть реализовано: удобное расположение компонентов на формах ввода параметров и снабжение их контекстной помощью; проверка введенных параметров на соответствие разрешенным диапазонам; описание функционала подключаемых библиотек в общем справочном разделе основной программы.

Хранение параметров и результатов экспериментов в базе данных. В базе данных нужно хранить все условия и результаты экспериментов, включая список кодеров и их параметры. Необходимо учитывать, что внешние модули, реализующие какие-либо алгоритмы, могут быть отключены или заменены.

Техническое решение проблемы подключения библиотек сторонних разработчиков состоит в абстрагировании основной программы от данных, используемых подключаемыми модулями, унификации требований к подключаемым библиотекам, снабжении каждой библиотеки специальным файлом описания и создании для каждой библиотеки процедуры редактирования введенных параметров. Рассмотрим элементы найденного решения подробнее.

Унифицированные требования к подключаемым библиотекам. Вне зависимости от реализуемых алгоритмов (кодирование и декодирование, перемежение и деперемежение или построение потока ошибок) ко всем библиотекам предъявляется ряд одних и тех же требований. Так все процедуры и функции внутри подключаемых библиотек, должны использовать соглашение cdecl для своего вызова. Каждая функция обязана хранить данные для своей работы (параметры для алгоритмов и промежуточные данные) в динамически выделяемой области памяти. Взаимодействие с основной программой основано на том, что ИС «Канал» манипулирует указателем на эту область памяти, вне зависимости от хранящихся в ней данных. Для работы ИС «Канал» с этим указателем разработчикам подключаемых библиотек необходимо написать пять сервисных функций, выполняющих базовые функции:

- создание записи (как структуры данных) для хранения параметров функций, вызываемых из подключаемой библиотеки;
- удаление записи;
- создание дубликата записи (дубликат используется для быстрого построения плана эксперимента);
- пользовательский ввод параметров с проверкой корректности;
- получение текстового описания параметров модуля.

Так как между основными элементами модели канала данные передаются в виде файлов конечной длины, то функции кодирования и декодирования, перемежения и деперемежения, а также внесения шума должны иметь два обязательных параметра типа pchar и указывающих на названия входного и выходного файлов, далее при необходимости располагается указатель на запись с входными параметрами алгоритма.

Ввод параметров алгоритмов. Входные значения для функций из внешних библиотек устанавливаются пользователями ИС «Канал» при указании параметров проводимых экспериментов во время выполнения основного модуля программы. Так как число параметров, их типы и множество допустимых значений могут быть произвольными, то создавать специальные пользовательские формы редактирования параметров должны разработчики сторонних модулей. Им известен диапазон допустимых значений параметров и только они могут удобно разместить компоненты редактирования этих параметров на пользовательской форме, снабдив их сопутствующей справочной информацией.

Вид процедуры редактирования данных жестко регламентируется. Например, на языке Pascal: procedure ParamInput (AppHand:THandle; pData:pointer); cdecl; export. Параметры функции: AppHand — дескриптор главного окна приложения; pData — указатель на тип Record, содержащий параметры алгоритма. Процедура редактирования данных не завершает свое выполнение до тех пор, пока пользователь корректно не введет все необходимые параметры.

Файл описания подключаемого модуля. Для повышения дружелюбности интерфейса ИС «Канал» в части работы с библиотеками сторонних разработчиков каждая библиотека должна быть снабжена специальным файлом описания, название которого совпадает с названием библиотеки, и имеет расширение ini. Такие файлы формируют разработчики подключаемой библиотеки. Использование этих файлов позволяет предоставлять справочную информацию о подключаемых модулях, обеспечивать возможность редактирования пользователем названий и описаний подключаемых кодеков, перемежителей и шумов, которые затем будут отображаться в пользовательском интерфейсе ИС «Канал». Файл описания является текстовым, его редактирование возможно в любом редакторе с поддержкой кодировки Windows-1251 (например, в стандартной программе ОС Windows «Блокнот»). Все изменения, сделанные в файлах описания, вступают в силу после перезагрузки программы, реализующей ИС «Канал».

Файл описания может состоять из нескольких блоков, каждый из которых посвящен описанию одного реализуемого библиотекой алгоритма (кодека, перемежителя или генератора шума). На рис. 3 представлен пример одного блока файла описания для внешней библиотеки, реализующей алгоритмы генерации потоков ошибок. Комментарии в каждой строке помещены после символа «%». Первая строка блока описания содержит служебное название алгоритма, помещенное в квадратные скобки. Внутри файла это название должно быть уникальным и не может быть пустой строкой. Далее до следующего служебного названия алгоритма или до конца файла идут параметры этого алгоритма. Каждый параметр находится в отдельной строке, и представляет собой выражение вида: <зарезервированное слово> = <значение>. Параметр «Type» может принимать одно из трех значений: 1 — для кодеров и декодеров, 2 — для перемежителей и деперемежителей, 3 — для генератора шума. Параметр «Name» является строкой символов неограниченной длины, в него записывается имя алгоритма, которое будет отображаться в пользовательском интерфейсе ИС «Канал». «Hint» — строка-подсказка, которая отображается в программе, когда курсор несколько секунд находится над

списком алгоритмов. «Description» — содержит справочную информацию по алгоритму, длина этого описания не ограничена. Справочная информация отображается при выборе меню «Справка» в пользовательском интерфейсе ИС «Канал». «DataControlDll» — имя библиотеки, содержащей процедуру редактирования данных и форму для ввода параметров алгоритма пользователем. Если параметр «DataControlDll» пуст, то считается, что форма находится внутри модуля, которому соответствует файл описания. «EditData» — имя экспортируемой процедуры «редактирования данных». «UniqueID» — уникальный идентификатор алгоритма, который в файл описания вставляет информационная система. «CreateData» — имя сервисной функции создания новой записи с данными. «DuplicateData» — имя сервисной функции создания дубликата записи. «DescriptionData» — имя сервисной функции, которая получает текстовое описание параметров модуля. «DisposeData» — имя сервисной функции уничтожения записи с данными.

Остальные параметры, представляемые в файле описания, зависят от типа алгоритма, указанного в «Type». Так для генераторов шумов необходимо указать «FuncName» — имя экспортируемой функции генератора шумов из DLL-библиотеки. Для кодеков и перемежителей необходимо указать: «Coder» — имя экспортируемой из DLL-библиотеки функции кодирования или перемежения, «Decoder» — имя экспортируемой функции декодирования или деперемежения.

```
[GenerationNoise]                     %  Служебное название алгоритма
Type=3                                %  Тип алгоритма – генератор шума
Name=Равномерный шум                  %  Название алгоритма, которое будет видно
                                      %  в интерфейсе пользователя ИС «Канал»
Hint=Функция генерации простого шума           %  Контекстная подсказка
Description=В поток данных вносится равномерно распределенные двоичные ошибки с постоянной вероятно-
стью p.                               % Описание для файла-справки
FuncName=GenerationNoise              % Имя функции генерации шума
DataControlDll=datacontrol.dll        % Процедура редактирования данных
                                      % находится  в библиотеке datacontrol.dll
EditData=ShowForm                     % Название формы из datacontrol.dll для ввода
                                      % пользователем параметров алгоритма
CreateData=DataCreate                 % имя сервисной функции создания
                                      % новой записи с данными.
DuplicateData=DataDuplicate           % имя сервисной функции создания дубликата записи
DescriptionData=DataDescription       % имя сервисной функции получения текстового описания
                                      % параметров модуля
```

Рис. 3. Пример блока файла описания

Алгоритм подключения внешних модулей. При каждом запуске исполняемого файла ИС «Канал» происходит поиск подключаемых библиотек в специальной директории «plugins», которая размещается в папке с программой. Схематично алгоритм подключения библиотек можно представить в следующем виде:

1. Просматриваем файлы в папке plugins с подключаемыми библиотеками.
2. Если находим файл с расширением dll, для которого существует файл описания, т.е. файл с таким же именем и расширением ini, то читаем файл описания и получаем из него данные о находящихся внутри модуля кодеках, перемежителях, генераторах шумах и именах функций, реализующих эти алгоритмы. Кроме этого, из файла описания извлекаются имя функции ввода и контроля данных, имена сервисных функций и имя библиотеки, в которой они находятся.
3. Загружаем необходимые библиотеки в память.
4. Если в пунктах 2–3 не было ошибок, добавляем названия внешних функций, полученных из подключенных библиотек в специальный список, который формируется в основном модуле ИС «Канал» и который содержит названия функций доступных для дальнейшего использования
5. Переходим к следующему файлу в папке plugins.

Недостатки предложенного технического решения. Основываясь на опыте многократного использования ИС «Канал» как его авторами, так и сторонними разработчиками и исследователями, можно выделить два основных недостатка данной системы.

Во-первых, для стороннего разработчика появляются дополнительные сложности по написанию небольших вспомогательных функций. Однако их можно считать незначительными в сравнении с обращением к авторам программы за изменениями или с созданием остального функционала, реализованного в ИС «Канал». К тому же данный недостаток системы частично решается предоставлением шаблонов и готовых примеров для разработчиков.

Во-вторых, в данной системе затруднительным является построение сложных выборок из базы данных с результатами проведенных имитационных экспериментов. Так как основная программа ИС «Канал» манипулирует параметрами функций из внешних библиотек как с областью памяти и не выделяет отдельные элементы из списка параметров, то в базе данных параметры алгоритмов хранятся в виде длинной строки, что затрудняет сортировку результатов экспериментов по параметрам алгоритмов. Разделение строки параметров на части и организация в базе данных соответствующих полей, содержащих эти параметры, невозможны из-за большого многообразия таких параметров в различных алгоритмах. Данная проблема частично решается грамотным составлением плана экспериментов и стандартной в ИС «Канал» функцией построения графиков по заданным критериям.

Заключение. За время своего существования ИС «Канал» неоднократно успешно использовалась как в учебном процессе, так и в научно-исследовательских целях. Некоторые примеры использования информационной системы «Канал» можно найти в работах [2, 10, 11]. Считаем, что реализация механизма работы с модулями, используемая в «ИС Канал» получилась удачной. Этот механизм обладает практически всеми желаемыми свойствами, перечисленными выше. Положительные результаты использования ИС «Канал» подтвердили тот факт, что ее структура с точки зрения программной реализации выбрана грамотно и качественно реализована. Разработанный способ подключения модулей сторонних разработчиков в ситуации, когда параметры внешних функций заранее не известны, можно использовать и в других проектах.

Библиографический список

1. Деундяк, В. М. Методы оценки применимости помехоустойчивого кодирования в каналах связи / В. М. Деундяк, Н. С. Могилевская. — Ростов-на-Дону : Издательский центр ДГТУ, 2007. — 86 с.

2. Деундяк, В. М. Методы помехоустойчивой защиты данных / В. М. Деундяк, А. Э. Маевский, Н. С. Могилевская. — Ростов-на-Дону : Изд-во Южного федерального университета, 2014. — 309 с.

3. Имитатор цифрового канала передачи данных (ChannelSim) : свидетельство о государственной регистрации программ для ЭВМ / В. В. Золотарев, Г. В. Овечкин. — № 2005611304 ; дата регистрации 31.05.05.

4. Сергиенко, А. Б. Цифровая обработка сигналов / А. Б. Сергиенко. — Санкт-Петербург : Питер, 2011. — 758 с.

5. Giordano, A. Modeling of Digital Communication Systems Using Simulink / A. Giordano, A. Levesque // NY: John Wiley & Sons, Inc., 2015. — 380 p.

6. Информационная система «Канал» : свидетельство о государственной регистрации программ для ЭВМ / Н. С. Могилевская, К. А. Чугунный. — № 2008614602 ; дата регистрации 24.09.2008 г.

7. Могилевская, Н. С. О развитии информационной системы «Канал» / Н. С. Могилевская, К. А. Чугунный // Системный анализ, управление и обработка информации : сб. трудов 5-го междунар. семинара. — Ростов-на-Дону, 2014. — С. 411–417.

8. Могилевская, Н. С. Информационная система исследования эффективности алгебраических схем помехоустойчивой защиты в системах передачи данных [Электронный ресурс] / Н. С. Могилевская // Современные проблемы науки и образования. — 2015. — № 1. — Режим доступа : www.science-education.ru / 121–17127 (дата обращения : 16.02.2015).

9. Архангельский, А. Я. Программирование в Delphi для Windows / А. Я. Архангельский. — Москва : Бином-Пресс, 2010. — 1248 с.

10. Могилевская , Н. С. Экспериментальное исследование декодеров кодов Рида-Маллера второго порядка / Н. С. Могилевская, В. Р. Скоробогат, В. С. Чудаков // Вестник Дон. гос. тех. ун-та. — 2008. — Т. 8, № 3. — С. 231–237.

11. Могилевская, Н. С. Корректирующая способность декодера мягких решений троичных кодов Рида-Маллера второго порядка при большом числе ошибок / Н. С. Могилевская // Вестник Дон. гос. тех. ун-та. — 2015. — № 1. — С. 121–130.

References

1. Deundyak, V.M., Mogilevskaya, N.S. Metody otsenki primenimosti pomekhoustoychivogo kodirovaniya v kanalakh svyazi. [Methods for evaluating the applicability of error-correcting coding in communication channels.] Rostov-on-Don: DSTU Publ. Centre, 2007, 86 p. (in Russian).
2. Deundyak, V.M., Mayevskiy, A.E., Mogilevskaya, N.S. Metody pomekhoustoychivoy zashchity dannykh. [Methods of noise-immune data protection.] Rostov-on-Don: Izd-vo Yuzhnogo federal'nogo universiteta, 2014, 309 p. (in Russian).
3. Zolotarev, V.V., Ovechkin, G.V. Imitator tsifrovogo kanala peredachi dannykh (ChannelSim : svidetel'stvo o gosudarstvennoy registratsii programm dlya EVM. [Digital data channel simulator (ChannelSim)] Certificate of state registration of computer programs no. 2005611304, 2005 (in Russian).
4. Sergiyenko, A.B. Tsifrovaya obrabotka signalov. [Digital signal processing.] St. Petersburg: Piter, 2011, 758 p. (in Russian).
5. Giordano, A., Levesque, A. Modeling of Digital Communication Systems Using Simulink. NY: John Wiley & Sons, Inc., 2015, 380 p.
6. Mogilevskaya, N.S., Chugunniy, K.A. Informatsionnaya sistema «Kanal»: svidetel'stvo o gosudarstvennoy registratsii programm dlya EVM. [Information system “Channel”.] Certificate of state registration of computer programs no. 2008614602, 2008 (in Russian).
7. Mogilevskaya, N.S., Chugunniy, K.A. O razvitii informatsionnoy sistemy «Kanal». [On the development of the information system “Channel”.] Sistemnyy analiz, upravlenie i obrabotka informatsii: sb. trudov 5-go mezhdunar. seminara. [System analysis, management and information processing: Proc. 5th Int. Workshop.] Rostov-on-Don, 2014, pp. 411–417 (in Russian).
8. Mogilevskaya, N.S. Informatsionnaya sistema issledovaniya effektivnosti algebraicheskikh skhem pomekhoustoychivoy zashchity v sistemakh peredachi dannykh. [Information system research efficiency algebraic error correction scheme in data transmission systems.] Modern Problems of Science and Education, 2015, no. 1. Available at: www.science-education.ru (accessed: 16.02.2015) (in Russian).
9. Arkhangelskiy, A.Y. Programmirovanie v Delphi dlya Windows. [Programming in Delphi for Windows.] Moscow: Binom-Press, 2010, 1248 p. (in Russian).

Mogilevskaya, N.S., Skorobogat, V.R., Chudakov, V.S. Eksperimental'noe issledovanie dekoderov kodov Rida-Mallera vtorogo poryadka. [Experimental research of second-order Reed-Muller codes.] Vestnik of DSTU, 2008, vol. 8, no. 3, pp. 231–237 (in Russian).

10. Mogilevskaya, N.S. Korrektiruyushchaya sposobnost' dekodera myagkikh resheniy troichnykh kodov Rida-Mallera vtorogo poryadka pri bol'shom chisle oshibok. [Correcting capacity of soft-decision decoder of ternary Reed – Muller second-order codes with a large number of errors.] Vestnik of DSTU, 2015, no. 1, pp. 121–130 (in Russian).

Поступила в редакцию 21.04.2015
Сдана в редакцию 23.04.2015
Запланирована в номер 25.06.2015